How to do backpropagation in a brain

Geoffrey Hinton
Canadian Institute for Advanced Research
&
University of Toronto

What is wrong with back-propagation?

- It requires labeled training data. (fixed)
  - Almost all real data is unlabeled.
  - The brain needs to fit about 10^14 connection weights in only about 10^9 seconds. Labels cannot possibly provide enough information.
- The learning time does not scale well for many hidden layers. (fixed)
- The neurons need to send two different types of signal
  - Forward pass: signal = activity = y
  - Backward pass: signal = dE/dy

Training a deep network

- First train a layer of features that receive input directly from the pixels.

- Then treat the activations of the trained features as if they were pixels and learn features of features in a second hidden layer.

- Each time we add another layer of features we improve a bound on how well we are modeling the set of training images.

Discriminative fine-tuning

- First train multiple hidden layers greedily.
- Then connect some output units to the top layer of features and do backpropagation through all of the layers to fine-tune all of the feature detectors.
- On MNIST with no prior knowledge or extra data, this works much better than standard backpropagation and better than SVM's.

Using backpropagation for fine-tuning

- Greedily learning one layer at a time scales well to really big networks, especially if we have locality in each layer.
- We do not start backpropagation until we already have sensible weights that already do well at the task.
  - So the initial gradients are sensible and backpropagation only needs to perform a local search.
- Most of the information in the final weights comes from modeling the distribution of input vectors.
  - The precious information in the labels is only used for the final fine-tuning. It slightly modifies the features. It does not need to discover features.
  - So we can do very well when most of the training data is unlabelled.

But how can the brain back-propagate through a stack of RBM’s?

- Many neuroscientists think back-propagation is biologically implausible because they cannot see how neurons could possibly do it.
  - Chomsky used the same logic to infer that syntax is innate!
- Some very good researchers have postulated inefficient algorithms that use random perturbations.
  - Do you really believe that evolution could not find an efficient way to adapt a feature so that it is more useful to the higher-level features? (have faith!)

The idea

- Learning a stack of simple models, each of which is good at reconstructing its inputs, creates the conditions required to allow neurons to implement backpropagation in an elegant way.
  - The trick is to use temporal derivatives to represent error derivatives.
  - This allows the output of a neuron to represent an error derivative at the same time as it is also representing the presence or absence of a feature.
- This is a big assumption about cortical representations.
  - Is there any evidence for it? (velocity neurons?)

A prerequisite

- Consider a binary stochastic output unit , j, with a cross-entropy error function.

$$-\partial E / \partial x_j \quad = \quad t_j \; - \; p_j$$

derivative of the error w.r.t. The total input to j

target value

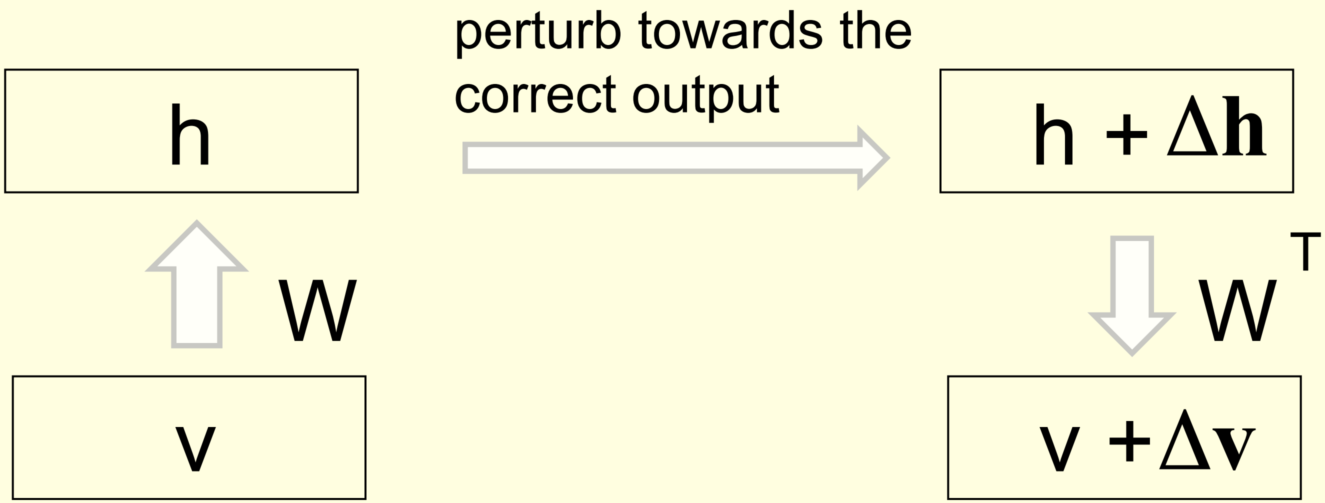
actual probability

So if we continuously regress the output probability towards the target value, we get

$$-\partial E / \partial x_j \quad \propto \quad \dot{p}_j$$

temporal derivative

Backpropagation is easy



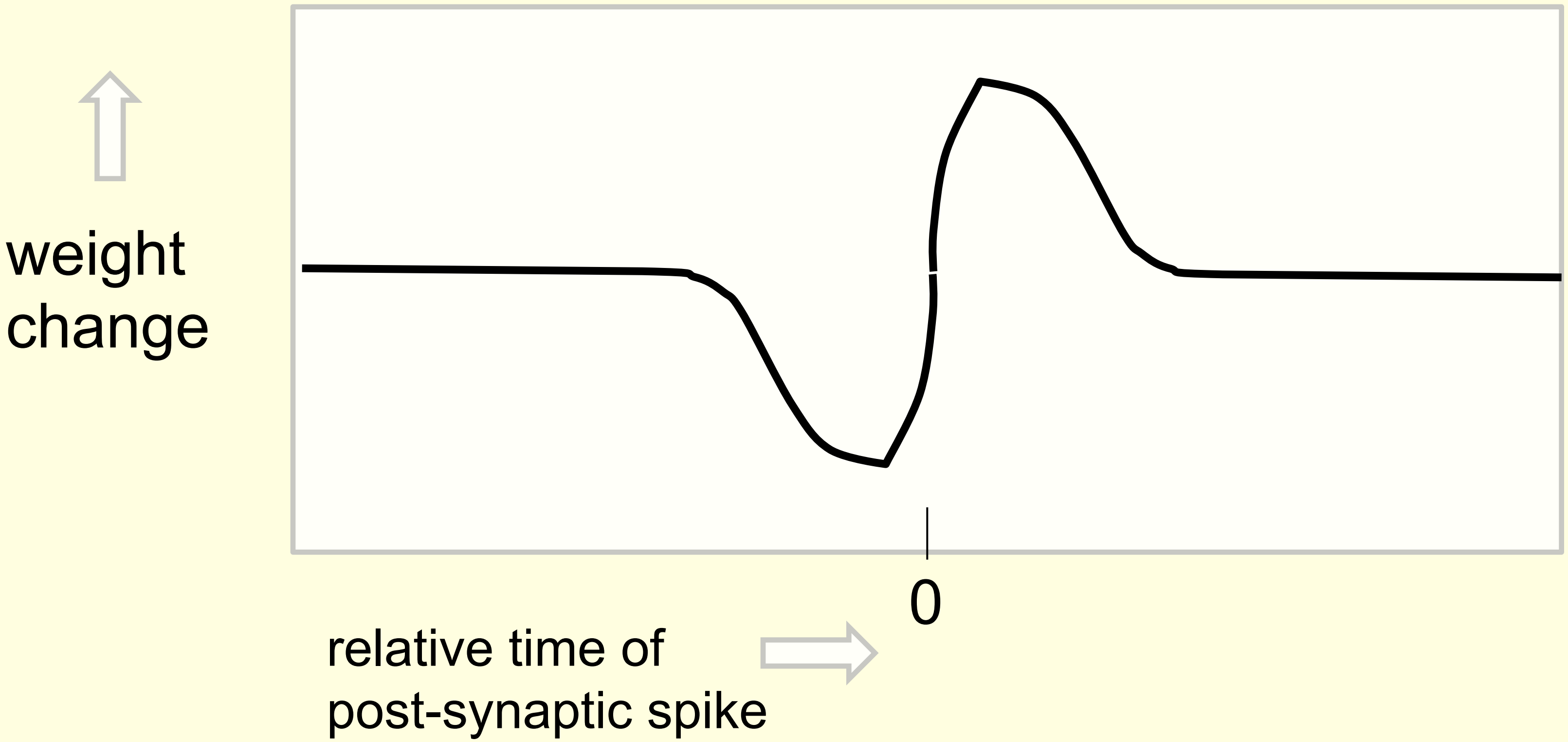
- Let component j of $\Delta \mathbf{h}$ represent $-\partial E / \partial x_j$ where x_j is the total input to neuron j
- If h would be reconstructed as v, h+$\Delta \mathbf{h}$ will be reconstructed as v+$\Delta \mathbf{v}$ where $\Delta \mathbf{v}$ is a vector with component i representing $-\partial E / \partial x_i$

The synaptic update rule

- First do a forward pass (as usual).
- Then perturb the top level activities so that the change in activity of a unit represents the derivative of the error function w.r.t. the total input to that unit.
- Then do a downwards pass.
  - This will make the activity changes at every level represent error derivatives.
- Then update each synapse in proportion to:

pre-synaptic activity X rate-of-change of post-synaptic activity

If this is what is happening, what should neuroscientists see?



- Spike-time-dependent plasticity is just a derivative filter.

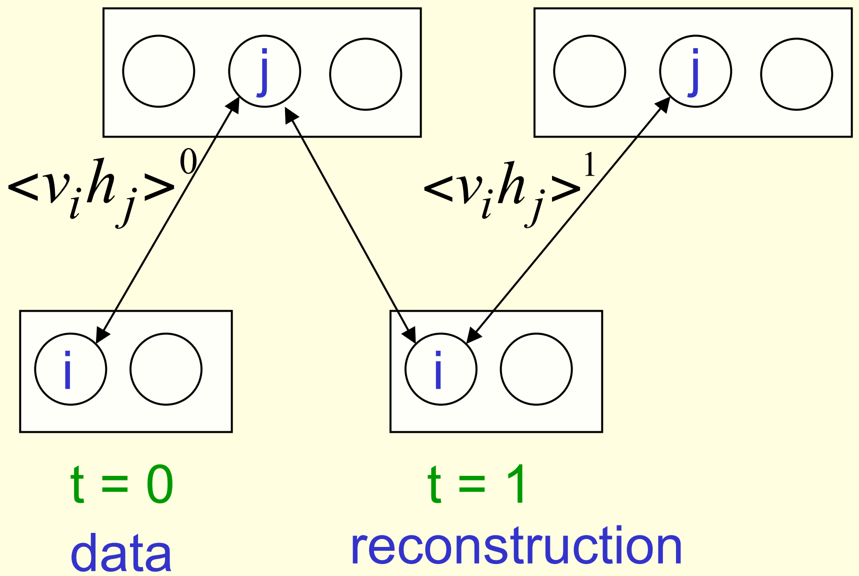
A problem

- This way of performing backpropagation requires symmetric weights
  – But contrastive divergence learning still works if we treat each symmetric connection as two directed connections and randomly remove many of the directed connections.

Functional symmetry

- The fluctuations in the hidden units are conditionally independent, but the state of a hidden unit can still be estimated very well from the states of other hidden units that have similar receptive fields.
  – So top-down connections from these other correlated units can learn to mimic the effect of the missing top-down part of a symmetric connection.
  – All we require is functional symmetry on and near the data manifold.
  – Contrastive divergence learning seems to achieve functional symmetry well enough to make backpropagation work.

Contrastive divergence learning: A quick way to learn an RBM



Start with a training vector on the visible units.

Update all the hidden units in parallel

Update all the visible units in parallel to get a "reconstruction".

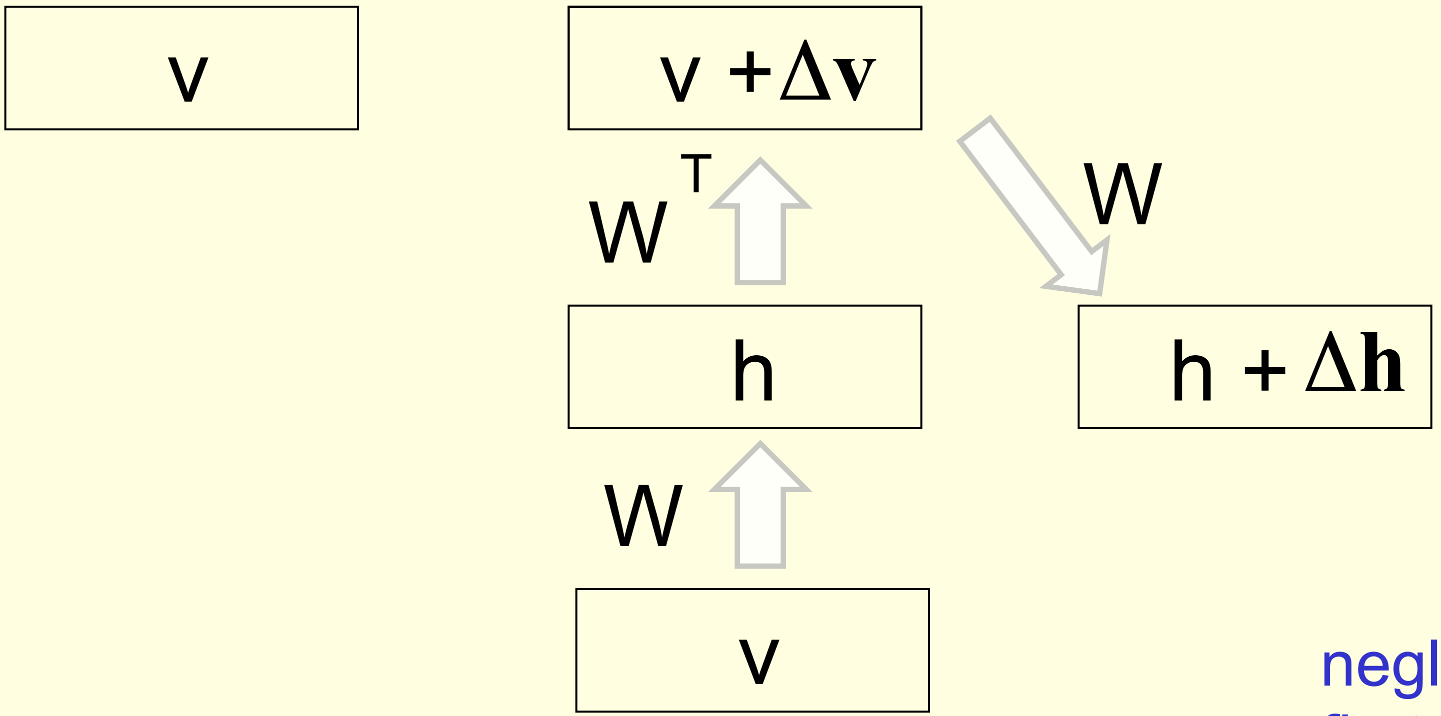
Update all the hidden units again.

$$\Delta w_{ij} = \varepsilon \left(<v_i h_j>^0 - <v_i h_j>^1 \right)$$

This is not following the gradient of the log likelihood. But it works well.

It is approximately following the gradient of another objective function.

Backpropagation learning of an autoencoder using temporal derivatives



negligible to first order

Backprop: $-\mathbf{v}\Delta\mathbf{h} - \mathbf{h}\Delta\mathbf{v}$

CD: $\mathbf{vh} - (\mathbf{v} + \Delta\mathbf{v})(\mathbf{h} + \Delta\mathbf{h}) = -\mathbf{v}\Delta\mathbf{h} - \mathbf{h}\Delta\mathbf{v} - \Delta\mathbf{v}\Delta\mathbf{h}$

THE END

And if you reserved a place on a bus to whistler, please make sure you get on the bus that you have been assigned to because the buses are all full.